A Universal Control Architecture for Maritime Cranes and Robots Using Genetic Algorithms as a Possible Mapping Approach

F. Sanfilippo [1], L. I. Hatledal [1], H. G. Schaathun [2], K. Y. Pettersen [3] and H. Zhang [1]

[1]Department of Maritime Technology and Operations (Avdeling for Maritim teknologi og Operasjoner), Aalesund University College (Høgskolen i Aalesund), Postboks 1517, 6025 Aalesund, Norway, [fisa, laht, hozh]@hials.no

[2]Department of Engineering and Natural Sciences (Avdeling for ingeniør- og realfag), Aalesund University College (Høgskolen i Aalesund), Postboks 1517, 6025 Aalesund, Norway, hasc@hials.no

[3]Department of Engineering Cybernetics, Norwegian University of Science and Technology, 7491 Trondheim, Norway, kristin.y.pettersen@itk.ntnu.no

This work is supported by the Innovation Programme for Maritime Activities and Offshore Operations (MAROFF) which is promoted by the Research Council of Norway

IEEE ROBIO 2013

Summary

Current maritime crane control architecture

Low control flexibility and non-standardisation are two crucial issues

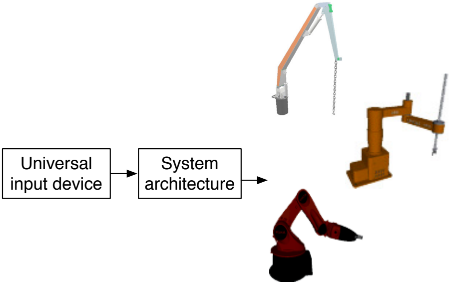
- relatively simple control interfaces
- array of levers, throttles or buttons are used to operate the crane joint by joint
- each input device can normally control only one specific crane model

When considering working efficiency and safety, this kind of control is extremely difficult to manage and extensive experience with high control skill levels is required of the operators

Currently used input devices

Underlying idea

A general architecture that allows for modelling, simulation and control of different models of maritime cranes and, more generally, robotic arms by using the same universal input device



Main challenges

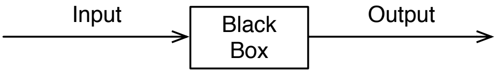
- mapping the fixed degrees of freedom of the universal input device to the variable degrees of freedom of the cranes or robots to be controlled
- designing and testing different mapping procedures (a case study method based on the use of Genetic Algorithms (GA) is presented)

Machine learning procedures

- A common assumption in the previous literature is that the IK model of the arm to be controlled is a priori knowledge
- This method is not very flexible, especially when planning to control different arms using a universal input device because several IK models are needed: one for each arm or crane to be controlled

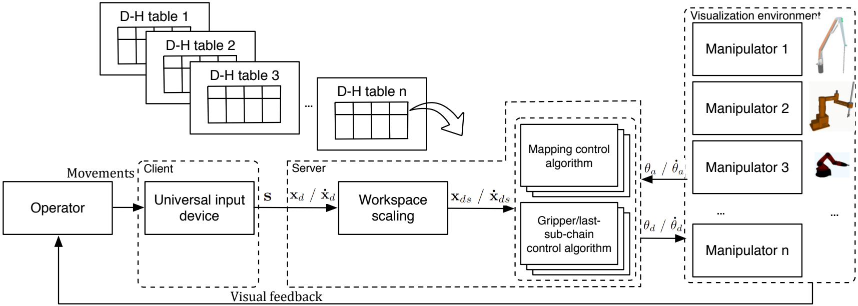
An alternative control method based on machine learning procedures

- Using methods that do not assume a priori knowledge for the IK model of the arm
- In this way the system would be able to automatically learn the kinematic properties of different arms and new models could also be easily added



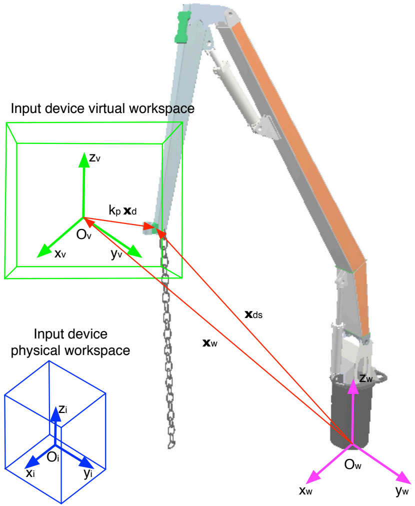
Most of the previous works in this field are only able to learn the control of a specific crane/arm. No common universal input device to control various cranes/arms with different kinematics exists

The proposed control system architecture, overview



The proposed architecture provides the possibility of controlling the arms in position mode or velocity mode

The proposed control system architecture, workspace scaling



The proposed architecture allows for expanding and shifting the small-scale physical workspace of the input device to a virtual expanded workspace allowing the robot arm for more accurate and precise movements.

$$\mathbf{x}_{ds} = k_p \mathbf{x}_d + \mathbf{x}_w, \qquad (1)$$

where k_p is the position scaling factor and $\mathbf{x}_w$ is a shifting vector that defines the position of the virtual reference frame with respect to the global reference frame.

$$\dot{\mathbf{x}}_{ds} = k_v \dot{\mathbf{x}}_d, \qquad (2)$$

where, k_v is the velocity scaling factor

The proposed control system architecture, mapping control algorithm

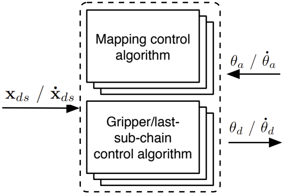
For all the different models to be controlled, the mapping methods have to implement the classic inverse kinematic functions that can be generalised as follows:

$$\theta_d = f_p^{-1}(\mathbf{x}_{ds}), \qquad (3)$$

concerning position control, and

$$\dot{\theta}_d = f_v^{-1}(\theta_a, \dot{\mathbf{x}}_{ds}), \qquad (4)$$

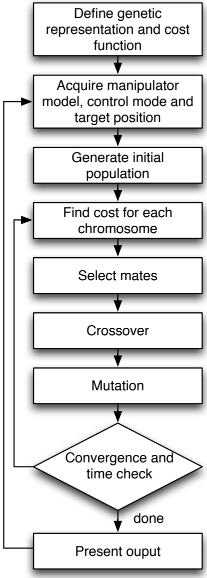
for velocity control, where θ_a is the the actual joint angles vector.



The proposed architecture also allows the end-effector to be modelled as a distinct sub-chain that can be controlled separately. In general, a mapping control method may or may not consider the control of the whole manipulator.

Flowchart

- A continuous GA is employed to automatically learn the mapping functions for the manipulators to be controlled. This approach only requires the forward kinematic model
- Note that the same set-up of the proposed algorithm is adopted independently of which manipulator is being controlled and whether the selected control mode is position or velocity



Define genetic representation and cost function

- Each chromosome encodes a set of joint angles, θ_g
- Fitness function:

$$d(\mathbf{x}_t, \mathbf{x}_a) = |\mathbf{x}_t - \mathbf{x}_a|, \quad (5)$$

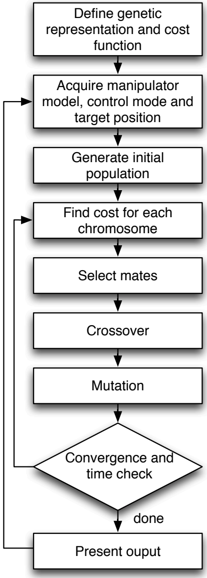
where, the actual position is calculated by using forward kinematics, while the target position depends on the input and it is given by:

$$\mathbf{x}_t = \mathbf{x}_{ds}, \quad (6)$$

if operating in position control mode, or by:

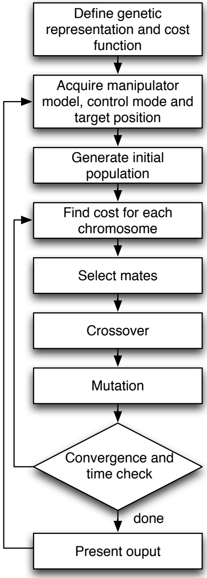
$$\mathbf{x}_t = \mathbf{x}_a + \dot{\mathbf{x}}_{ds}\Delta t, \quad (7)$$

if operating in velocity control mode, where Δt is the time interval between two successive iterations



Selection, crossover, mutation

- The selection is obtained by using the *stochastic universal sampling method*, which is a fitness proportionate selection method
- The crossover function is defined as a hybrid function that stochastically switches - with a 50% crossover probability - between using a single-point and a uniform crossover method
- mutation may occur by stochastically adding a random value of $\pm 5\%$. In particular, there is a 0.5% mutation chance for each gene. Additionally, a form of elitism is also used and 10% of the fittest chromosomes survives unaltered between generations



Convergence, time check and output presentation

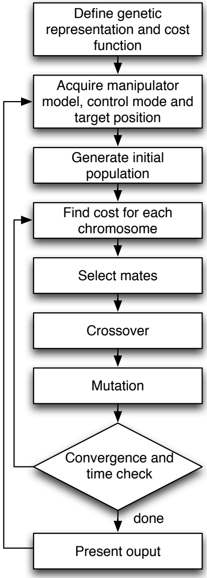
- The GA population stops evolving and the fittest chromosome is returned when the cost drops below 0.01 or when the overall time spent evolving the population exceeds 100 ms
- Denoting the fittest chromosome as θ_f and according to the operation scenario, the output is obtained as:

$$\theta_d = \theta_f, \qquad (8)$$

when operating in position control mode, or as:

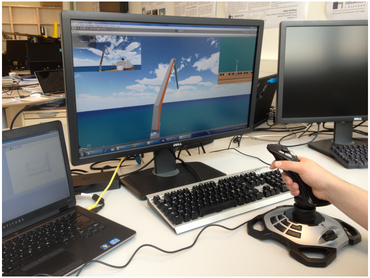
$$\dot{\theta}_d = \frac{\theta_f - \theta_a}{\Delta t}, \qquad (9)$$

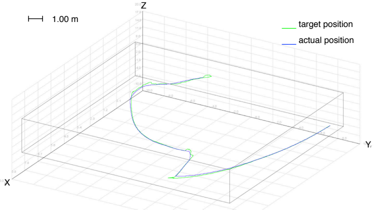
when operating in velocity control mode



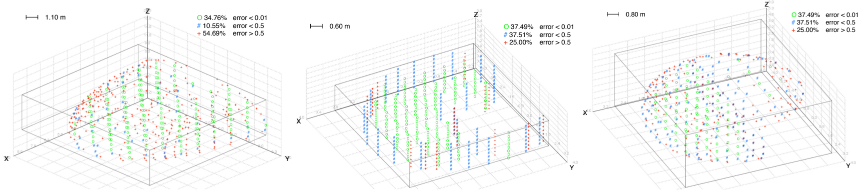
Simulations and Experimental Results

- The system is based on a distributed structure and the communication between client, server and visualisation environment is realised by using the *TCP/IP* protocol. This makes it also possible to remotely control the different manipulators
- Related simulations were carried out in order to test the architecture within the particular case study of the proposed mapping method. A knuckle boom crane, a *SCARA* robot and a *KUKA youBot* robot are modelled and simulated





Simulations and Experimental Results



3D Scatter plots showing error distribution for 512 equally-spaced target positions in the volume box that encloses the workspace the knuckle boom crane model, the *SCARA* robot and the *KUKA youBot* robot.

Future work

- The adopted cost function that is currently related to the workspace of the controlled model, could be also be related, as future work, to the particular manipulation task to be performed
- New mapping methods which also take heave compensation and swing related problems into account could also be implemented and integrated allowing for better flexibility and reliability of the proposed framework
- As next state of development, the proposed system architecture could be used for controlling real industrial robots or real maritime cranes

Ongoing work

Thank you for your attention